Java
Rest Services
OpenAPI Swagger

JEE Microservices

@ CGS IT – 2023

Version 1.0.5

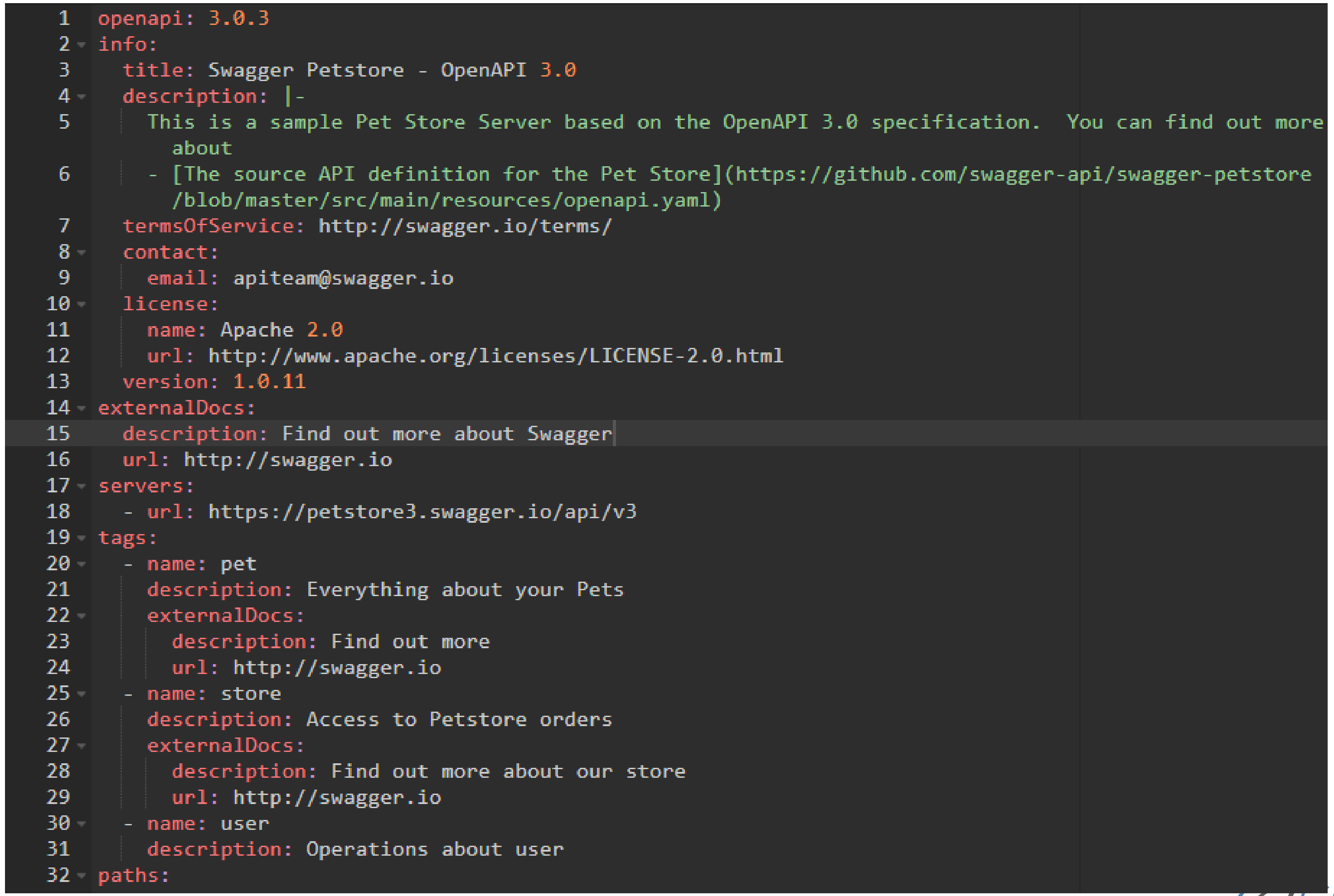
Inhalt

Open API - Spezifikation

- The OpenAPI Specification (OAS) defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection.
- When properly defined, a consumer can understand and interact with the remote service with a minimal amount of implementation logic.
- An OpenAPI definition can then be used by documentation generation tools to display the API,
- Code generation tools to generate servers and clients in various programming languages, testing tools, and many other use cases.

Open API – Document Structure

- OpenAPI : versions info
- Info Sektion für generelle Infos



```yaml
openapi: 3.0.3
info:
  title: Swagger Petstore - OpenAPI 3.0
  description: |-
    This is a sample Pet Store Server based on the OpenAPI 3.0 specification.  You can find out more about
    - [The source API definition for the Pet Store](https://github.com/swagger-api/swagger-petstore/blob/master/src/main/resources/openapi.yaml)
  termsOfService: http://swagger.io/terms/
  contact:
    email: apiteam@swagger.io
  license:
    name: Apache 2.0
    url: http://www.apache.org/licenses/LICENSE-2.0.html
  version: 1.0.11
externalDocs:
  description: Find out more about Swagger
  url: http://swagger.io
servers:
  - url: https://petstore3.swagger.io/api/v3
tags:
  - name: pet
    description: Everything about your Pets
    externalDocs:
      description: Find out more
      url: http://swagger.io
  - name: store
    description: Access to Petstore orders
    externalDocs:
      description: Find out more about our store
      url: http://swagger.io
  - name: user
    description: Operations about user
paths:
```

CGS ITSolutions

Info Annotations

```
@OpenAPIDefinition(
    tags = {
        @Tag(name = "widget", description = "Widget operations."),
        @Tag(name = "gasket", description = "Operations related to gaskets")
    },
    info = @Info(
        title = "Chat Message Example API",
        version = "1.0.1",
        contact = @Contact(
            name = "Chat Message Example API Support",
            url = "http://exampleurl.com/contact",
            email = "techsupport@example.com"),
        license = @License(
            name = "Apache 2.0",
            url = "https://www.apache.org/licenses/LICENSE-2.0.html"))
)
public class ChatApplication extends Application {
}
```

Open API

Field Name	Type	Description
openapi	string	REQUIRED. This string MUST be the semantic version number of the OpenAPI
info	Info Object	REQUIRED. Provides metadata about the API. The metadata MAY be used by tooling as required.
Servers	[Server Object]	An array of Server Objects, which provide connectivity information to a target server. If the servers property is not provided, or is an empty array, the default value would be a Server Object with a url value of /.
Paths	Paths Object	REQUIRED. The available paths and operations for the API.
Components	Components Object	An element to hold various schemas for the specification.
security	[Security Requirement Object	A declaration of which security mechanisms can be used across the API.
tags	[Tag Object]	A list of tags used by the specification with additional metadata
externalDocs	External Documentation Object	Additional external documentation.

Siehe dazu: https://swagger.io/specification/v3/

OpenAPI – Maven Dependency

```xml
<dependency>
    <groupId>io.quarkus</groupId>
    <artifactId>quarkus-smallrye-
openapi</artifactId>
</dependency>
```

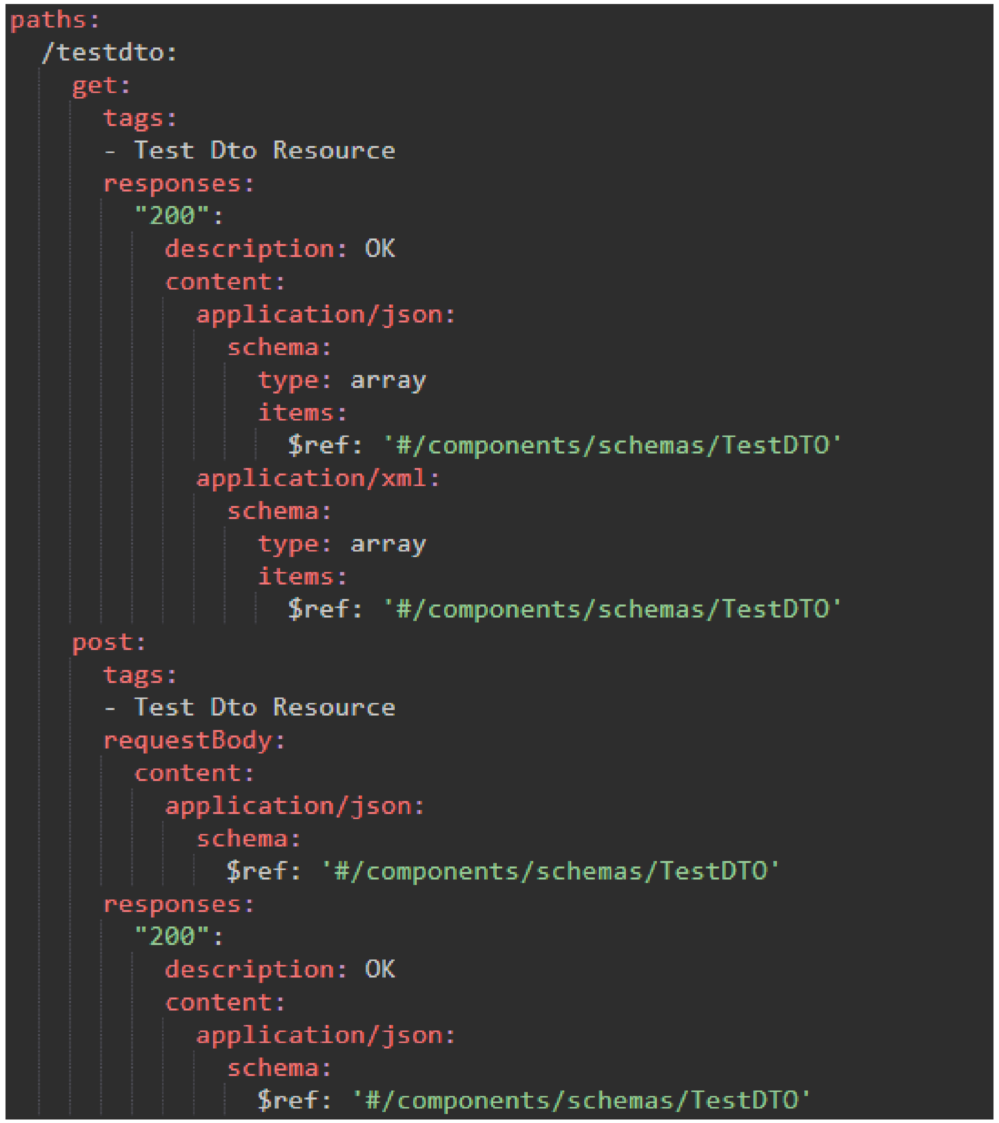
OpenAPI – Paths Beispiel

```
@GET
@Path("/")
@Produces({MediaType.APPLICATION_JSON,
MediaType.APPLICATION_XML})
public List<TestDTO> listAllObects(){
```

OpenAPI – Paths Beispiel

- Für das Test DTO werden die einzelnen implementieren Methoden folgendermaßen beschrieben:

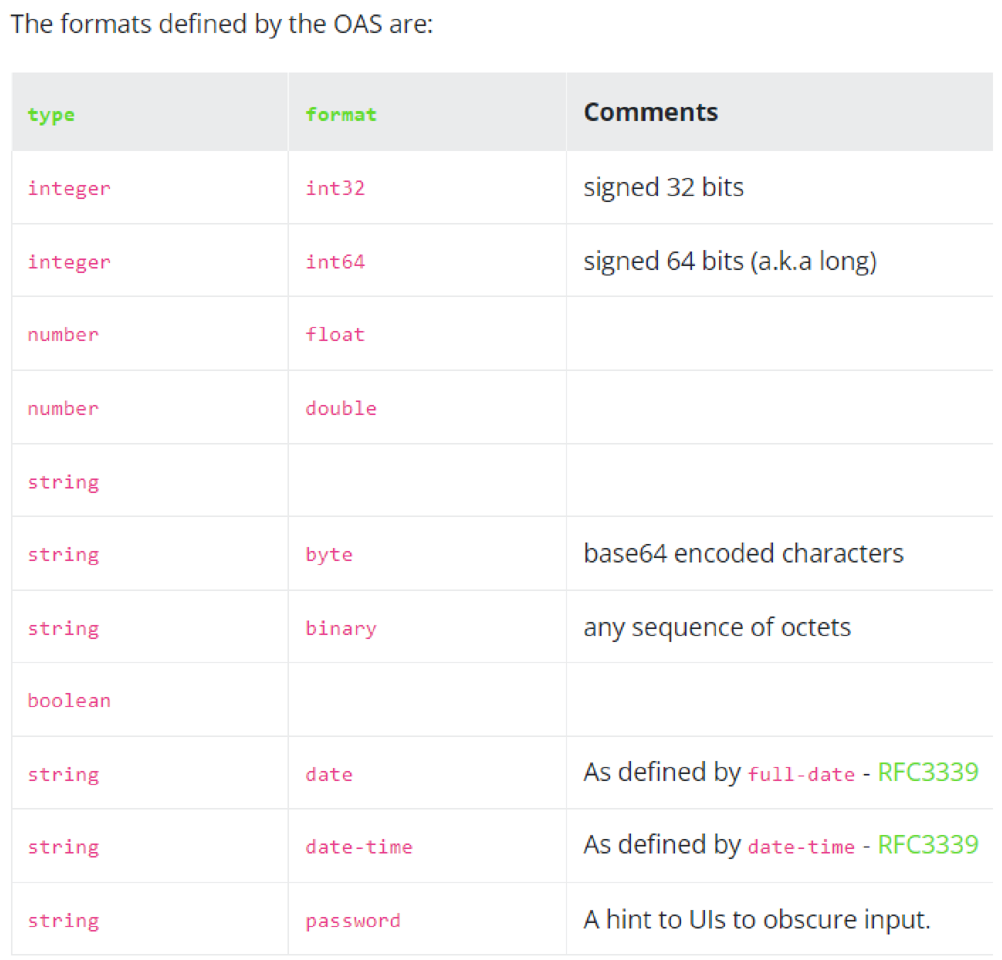
1. /testdto: Definiert den Basis Pfad für diese Beschreibung des TestDTO APIs
2. Get: wird hier ohne weiteren Pfad dokumentiert für die http GET Aufruf
3. Für den GET Aufruf ist bisher nur eine Antwort „200“ als OK dokumentiert
4. Es könnten aber auch weitere Fehlerfälle mit anderen http Error Codes beschrieben werden
5. Als Content für den Response werden sowohl eine JSON als auch eine XML Version des DTOs beschrieben.
6. Die Antwort ist wie im Java Code Angegeben eine Liste, die hier als Array abgebildet ist.
7. Die Elemente des Arrays werden hier als Items beschreiben und mittels $ref in als DTO



```yaml
paths:
  /testdto:
    get:
      tags:
      - Test Dto Resource
      responses:
        "200":
          description: OK
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/TestDTO'
            application/xml:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/TestDTO'
    post:
      tags:
      - Test Dto Resource
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/TestDTO'
      responses:
        "200":
          description: OK
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/TestDTO'
```

CGS ITSolutions

OpenAPI – DataTypes



The formats defined by the OAS are:

type	format	Comments
integer	int32	signed 32 bits
integer	int64	signed 64 bits (a.k.a long)
number	float	
number	double	
string		
string	byte	base64 encoded characters
string	binary	any sequence of octets
boolean		
string	date	As defined by full-date - RFC3339
string	date-time	As defined by date-time - RFC3339
string	password	A hint to UIs to obscure input.

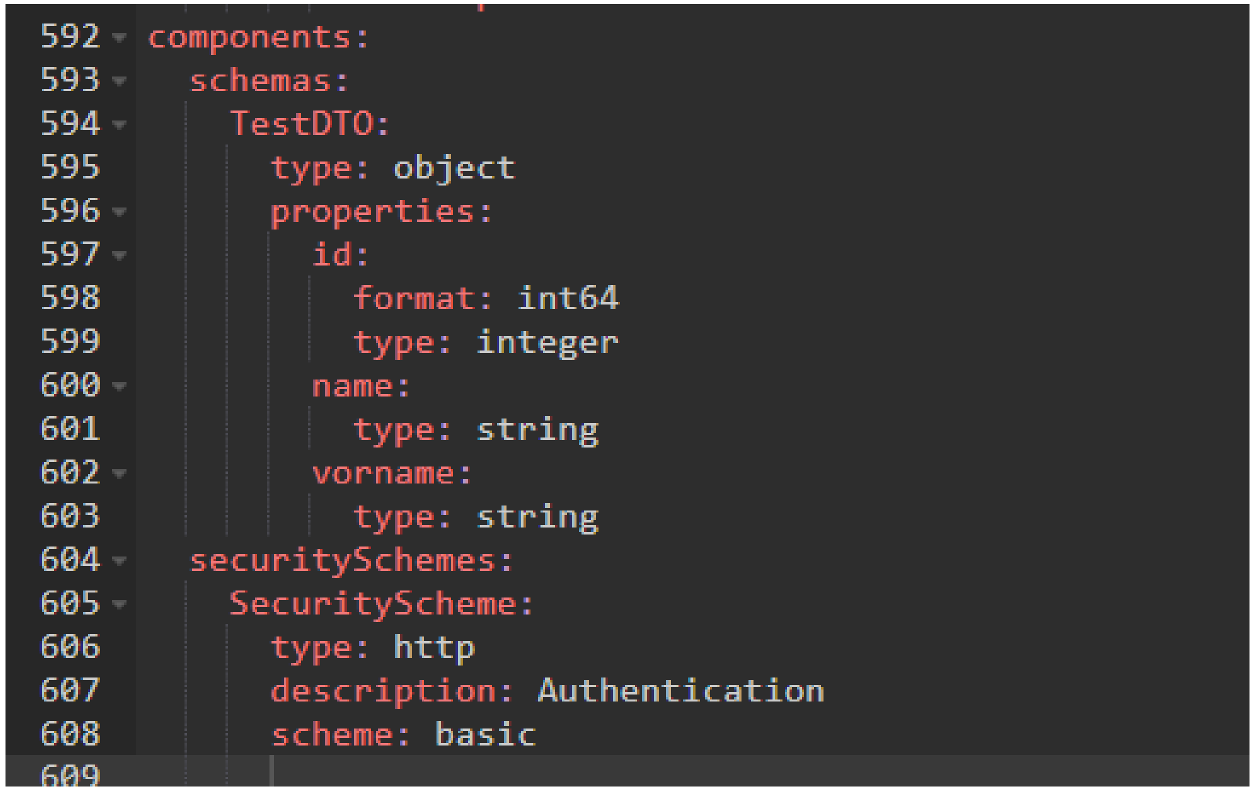
- Primitive data types in the OAS are based on the types supported by the JSON Schema Specification Wright Draft 00.
- Primitives have an optional modifier property: format. OAS uses several known formats to define in fine detail the data type being used.

CGS ITSolutions

OpenAPI – DTO Abbildung

```java
public class TestDTO {

    private Long id;
    String name;
    String vorname;
```



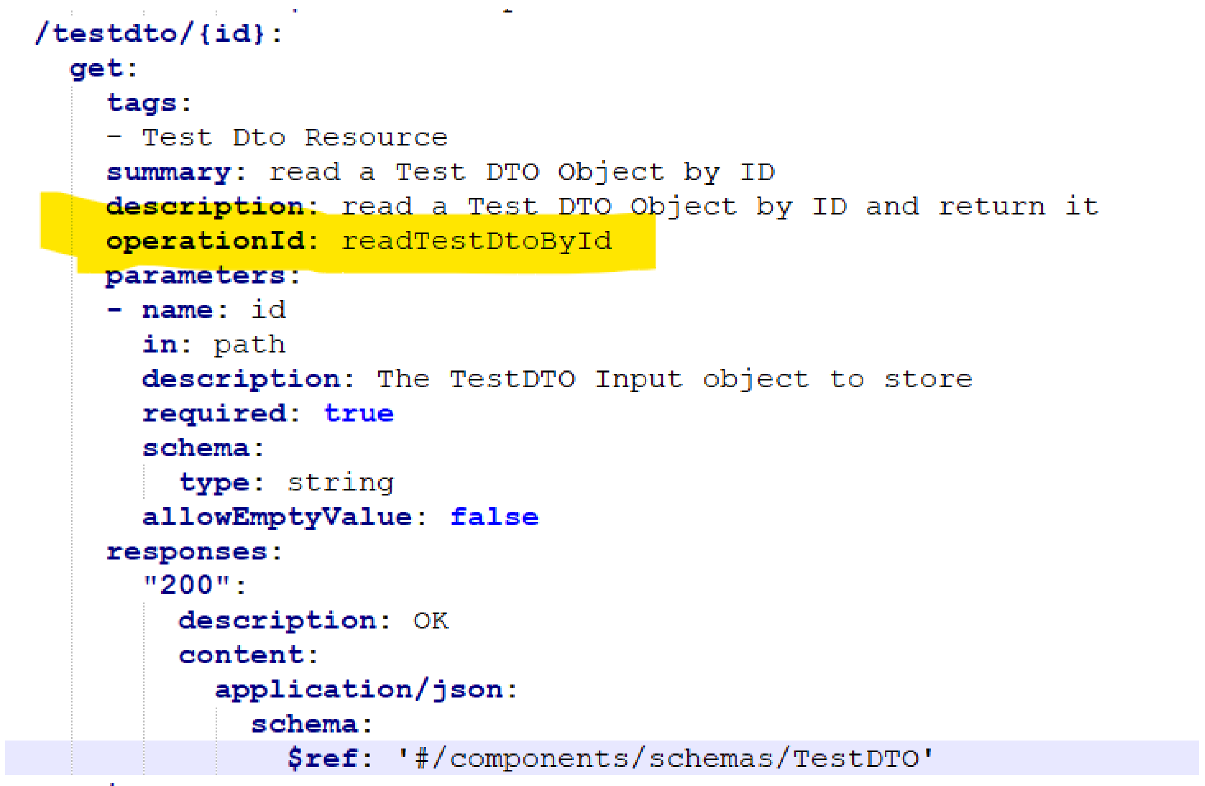
```yaml
components:
  schemas:
    TestDTO:
      type: object
      properties:
        id:
          format: int64
          type: integer
        name:
          type: string
        vorname:
          type: string
  securitySchemes:
    SecurityScheme:
      type: http
      description: Authentication
      scheme: basic
```

OpenAPI – Method OperationID

- Unique (UseCase) Identifier für die Operations im API
- If provided, these IDs must be unique among all operations described in your API.

```
@Operation( summary = "read a Test DTO Object by ID",
     description = "read a Test DTO..",
     operationId = "readTestDtoById")
@GET
@Path("/{id}")
@Produces(MediaType.APPLICATION_JSON)
public TestDTO readObjectById(@PathParam("id") String id
){

   log.infov("input {} , objectOutput {0}",  id, "");
```



```
/testdto/{id}:
  get:
    tags:
    - Test Dto Resource
    summary: read a Test DTO Object by ID
    description: read a Test DTO Object by ID and return it
    operationId: readTestDtoById
    parameters:
    - name: id
      in: path
      description: The TestDTO Input object to store
      required: true
      schema:
        type: string
      allowEmptyValue: false
    responses:
      "200":
        description: OK
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/TestDTO'
```

OpenAPI – Query Parameter

- Query string parameters must not be included in paths. They should be defined as query parameters instead.

```
//
http://localhost:8080/parameter/queryParameter?qp=inputTex
t&qp2=text2
@GET
@Path("/queryParameter")
@Produces(MediaType.TEXT_PLAIN)
public String queryParameter(
        @QueryParam("qp") String qp,
        @DefaultValue("1") @QueryParam("qp2") Long qP2
        ){
    log.infov("log QueryParam: {0}", qp);
    return "query params ["+ qp + "] und [" + qP2 + "]";
}
```

```
/parameter/queryParameter:
  get:
    tags:
    - Parameter Resource
    parameters:
    - name: qp
      in: query
      schema:
        type: string
    - name: qp2
      in: query
      schema:
        format: int64
        default: "1"
        type: integer
    responses:
```

OpenAPI – Path Parameter

- Path Parameter werden mit ihrem Platzhalter im Path angegeben und als Parameter dokumentiert:

```
@GET
@Path("/inputParameter/{inputString}")
@Produces(MediaType.TEXT_PLAIN)
public String inputParameter(
        @PathParam("inputString") String inputString){
    log.infov("log: {0}", inputString);

    StringBuilder sbStr = new StringBuilder();
    sbStr.append(inputString).reverse();
    return sbStr.toString() + "{}";
}
```

```
/parameter/inputParameter/{inputString}:
  get:
    tags:
    - Parameter Resource
    parameters:
    - name: inputString
      in: path
      required: true
      schema:
        type: string
    responses:
      "200":
        description: OK
        content:
          text/plain:
            schema:
              type: string
```

Danke für Ihre Aufmerksamkeit

CGS
ITSolutions